Leonardo Ferreira Carneiro
leo@cbpf.br

Nilton Alves Júnior
naj@cbpf.br
http:/mesonpi.cat.cbpf.br/naj

Resumo

Esse trabalho tem como objetivo abordar determinados aspectos relativos a roteadores e segurança em redes, com alguns exemplos típicos. Ao longo dessa Nota Técnica, serão vistos um pequeno histórico sobre a evolução dos sistemas de computação, os conceitos de roteadores, *firewalls*, *access lists* e segurança em redes. Dessa forma, pretendemos oferecer uma primeira visão à respeito desses assuntos.

Palavras-chave:

Roteador; Segurança em Redes; Comandos *Access List* e *Access Group*.

Índice

Roteadores e Segurança em Redes

1. Histórico

A comunicação sempre foi uma das maiores necessidades da sociedade humana. De acordo com o crescimento das civilizações, que ocupavam áreas cada vez mais dispersas geograficamente, a comunicação a longa distância se tornava uma necessidade cada vez maior e um desafio. Formas de comunicação através de sinais de fumaça ou pombos-correio foram as maneiras encontradas por nossos ancestrais para tentar aproximar as comunidades distantes.

Ao inventar o telégrafo em 1838, Samuel F. B. Morse inaugurou uma nova época nas comunicações. Nos primeiros telégrafos utilizados no século XIX, mensagens eram codificadas em cadeias de símbolos binários (código Morse) e então eram transmitidas manualmente por um operador através de um dispositivo gerador de pulsos elétricos. A partir daí, a comunicação através de sinais elétricos atravessou uma grande evolução, dando origem à maior parte dos grandes sistemas de comunicação encontrados atualmente, como o telefone, o rádio e a televisão.

Essa evolução no tratamento de informações não aconteceu somente na área de comunicação. Equipamentos para processamento e armazenamento de informações também foram alvo de grandes investimentos ao longo do nosso desenvolvimento. A introdução de redes de computadores na década de cinqüenta foi, provavelmente, o maior avanço do século neste sentido.

A união destas duas tecnologias – comunicação e processamento de informações – veio revolucionar o mundo de hoje, abrindo as fronteiras com novas formas de comunicação, e permitindo assim maior eficácia dos sistemas computacionais. Tais sistemas sofreram uma grande evolução desde o seu início, no Pós-Guerra, até os dias de hoje.

Embora a indústria de computadores seja jovem quando comparada com indústrias como a automotiva e a de transportes aéreos, os computadores tem feito um fantástico progresso em um curto espaço de tempo. Durante as suas duas primeiras décadas de existência, os sistemas de computação eram altamente centralizados, em geral, em uma única sala grande, sendo o computador uma máquina grande e complexa, operada por pessoas altamente especializadas. A noção de que dentro de vinte anos computadores igualmente poderosos, consideravelmente menores, pudessem ser produzidos em massa era considerada inviável.

No entanto, nos anos setenta, a introdução dos PC's revolucionou esses sistemas de computação, substituindo o modelo de um único computador servindo a todas as necessidades computacionais de uma organização por outro no qual um grande número de computadores separados, mas interconectados, executam essa tarefa. Através dessa distribuição de poder computacional, chegou-se então as arquiteturas de redes de computadores que encontramos hoje em dia.

Contudo, uma única rede local está sujeita a certos limites, como por exemplo o número de estações que podem ser conectadas a ela, a velocidade na transmissão dos dados entre as estações ou ainda quanto tráfego a rede pode suportar. Para superar essas limitações, surgiram, a partir dos anos oitenta, as chamadas *internetworks*. *Internetworking* é a ciência de interligar LAN´s individuais para criar WAN´s, e de conectar WAN´s para criar WAN´s ainda maiores. Uma LAN (*Local Area Network*) é uma rede de computadores que abrange uma área relativamente pequena,

enquanto uma WAN (*Wide Area Network*) é uma rede que ocupa uma maior área geográfica, consistindo geralmente de duas ou mais LAN's. Essas ligações inter-redes são executadas por dispositivos específicos, como por exemplo os roteadores, que são o assunto do próximo item [1][2][18].

2. Roteadores

Um roteador é um dispositivo que provê a comunicação entre duas ou mais LAN's, gerencia o tráfego de uma rede local e controla o acesso aos seus dados, de acordo com as determinações do administrador da rede. O roteador pode ser uma máquina dedicada, sendo um equipamento de rede específico para funções de roteamento; ou pode ser também um software instalado em um computador.

Consideremos por exemplo um grupo de dispositivos de rede, como servidores, PC's e impressoras, formando uma rede local a qual chamamos de LAN 1, como mostrado na figura 1. Consideremos também outra rede local, similar a primeira, a qual chamamos de LAN 2. A interconexão entre elas, que permite a troca de dados e o compartilhamento dos seus recursos e serviços, é feita pelo roteador. Esse esquema caracteriza o uso de uma máquina dedicada.
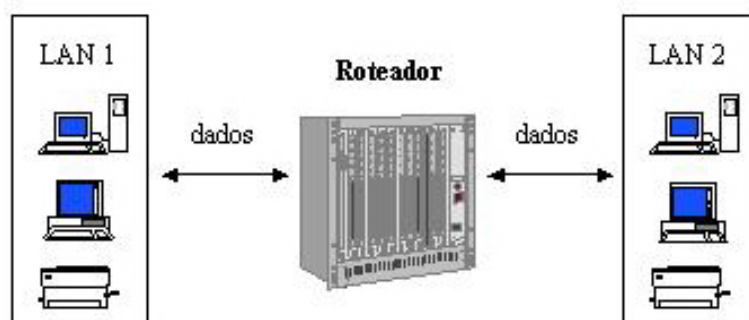
Figura 1: O roteador permite o tráfego de informações e o compartilhamento de serviços e recursos entre redes diferentes.

Consideremos agora a figura mostrada a seguir. Nela está representado o caso em que o roteador é um software instalado em um computador. Como podemos observar, é o computador, através de um software específico, que gerencia o tráfego de dados entres as diferentes redes mostradas. Esse esquema representa a topologia de rede inicialmente utilizada no CBPF até 1996, quando um servidor *Novell* exercia a função de um roteador, através de um software de roteamento fabricado pela própria *Novell*.

O roteador opera na camada de rede, a terceira das sete camadas do modelo de referência ISO OSI. Esse modelo de rede foi criado pela ISO (*International Organization of Standardization*) no início dos anos oitenta, tendo sido o primeiro passo para a padronização internacional dos diversos protocolos de comunicação existentes atualmente [2][5]. Para aqueles que estiverem interessados, maiores informações à respeito do modelo OSI podem ser encontradas na Nota Técnica intitulada "Redes de Computadores", NT – 008/98, de Outubro de 1998.

Quanto ao funcionamento de um roteador, temos que quando pacotes (partes da mensagem que é transmitida) são transmitidos de um *host* (qualquer dispositivo de uma rede) para outro, esses equipamentos usam cabeçalhos (*headers*) e uma tabela de roteamento para determinar

por qual caminho esses pacotes irão; os roteadores também usam o protocolo ICMP (*Internet Control Message Protocol*) para comunicarem-se entre si e configurarem a melhor rota entre dois *hosts* quaisquer. O cabeçalho, em várias disciplinas da ciência da computação, é definido como uma unidade de informação que antecede o objeto de dados de um pacote; ou seja, é no cabeçalho que está contida a informação sobre o destino do pacote utilizada pelo roteador. Já em relação ao ICMP, temos que ele é uma extensão do protocolo IP (*Internet Protocol*), sendo definido pela RFC 792. O ICMP suporta pacotes que contenham mensagens de erro, de controle e de informação. O comando *ping*, por exemplo, usa esse protocolo para testar uma conexão Internet [5][7][8].
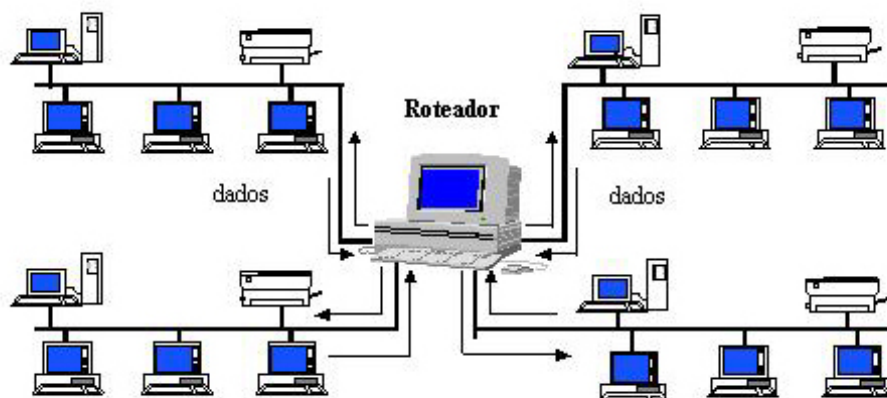
Figura 2: Um computador, através de software específico, pode gerenciar o tráfego de dados entre redes diferentes, funcionando como um roteador.

Por último, temos que uma pequena filtragem de dados é feita através de roteadores. Contudo, é importante ressaltar que os roteadores não se preocupam com o conteúdo dos pacotes com que eles lidam, verificando apenas o cabeçalho de cada mensagem, podendo ou não tratá-la de forma diferenciada [5].

2.1. Endereços IP e Redes TCP/IP

Ao longo dessa Nota Técnica, os termos **endereço IP** e **rede TCP/IP** serão amplamente utilizados. Devido a isso, esse item será dedicado exclusivamente a esses dois assuntos, de forma à oferecer uma melhor compreensão dos próximos itens.

2.1.1. Endereçamento IP

Um endereço IP é definido como sendo uma identificação para um computador ou um dispositivo qualquer de uma rede TCP/IP, que será explicada no próximo item.

Esses tipos de redes roteam mensagens baseadas no endereço IP de destino. O formato de um endereço IP é o de um endereço numérico de 32 bits escritos como 4 números, também conhecidos como octetos, que são separados por pontos, como por exemplo **152.84.253.47**. Cada um desses 4 octetos representam campos de 8 bits.

Com uma rede isolada, pode-se atribuir um endereço IP qualquer, respeitando o fato de que cada endereço deve ser único. Entretanto, o registro de uma rede privada na Internet requer endereços IP registrados, chamados endereços da Internet, para evitar possíveis duplicações.

Os 4 números ou octetos de um endereço IP são usados de maneiras diferentes para identificar uma rede particular e um *host* qualquer nessa rede. Classificam-se endereços da Internet registrados em 4 classes, listadas abaixo:

Classe A: Suporta 16 milhões de *hosts* em cada uma das suas 127 redes. Nessa classe de rede, temos que se o primeiro bit do seu endereço IP for 0, então os próximos 7 bits serão destinados ao número de rede e os 24 bits (3 octetos) restantes, aos números de dispositivo.

Classe B: Suporta 65.000 *hosts* em cada uma das suas 16.000 redes. Aqui, temos que se os 2 primeiros bits forem 1 e 0, respectivamente, então os próximos 14 bits serão destinados ao número da rede e os 16 bits (2 octetos) restantes aos números de dispositivos.

Classe C: Suporta 254 *hosts* em cada um dos seus 2 milhões de redes. Se os seus 3 primeiros bits forem 1, 1 e 0, respectivamente, então os próximos 21 bits serão destinados ao número de rede e os 8 bits (1 octeto) restantes aos números de dispositivos.

Classe D: Se os quatro primeiros bits forem 1, 1, 1 e 0, respectivamente, então o valor do primeiro octeto pode variar entre 224 e 239 e dizemos que esse número é um endereço *multicast*. Os próximos 28 bits compõem um número de identificação de grupo para um específico grupo *multicast*. Podemos concluir então que um endereço IP *multicast* é um endereço destinado a um ou mais *hosts* ou dispositivos, ao contrário dos endereços classe A, B e C, que especificam o endereço de um *host* ou dispositivo individual [2].

2.1.2. Redes TCP/IP

Uma rede de computadores é chamada de rede TCP/IP (*Transmission Control Protocol / Internet Protocol*) quando a sua arquitetura é baseada nesse conjunto de protocolos de comunicação. Esse tipo de rede baseia-se principalmente em: um serviço de transporte orientado à conexão, fornecido pelo protocolo TCP, e em um serviço de rede não-orientado à conexão, fornecido pelo protocolo IP. As informações enviadas pela Internet são dependentes do TCP/IP, fazendo com que ele seja utilizado como um protocolo primário de rede na Internet [1][2]. O roteamento de pacotes feito entre redes TCP/IP na Internet é o assunto do próximo item.

2.2. Roteamento na Internet

O roteamento é a técnica através da qual os dados encontram o seu caminho de um *host* para outro. No contexto da Internet, existem três aspectos principais de roteamento:

- ✓ Determinação do endereço físico.
- ✓ Seleção de roteadores (*gateways*) inter-redes.
- ✓ Endereços simbólicos e numéricos.

O primeiro desses três aspectos é necessário quando dados de uma rede TCP/IP estão para ser transmitidos por um computador. É necessário que esses dados sejam encapsulados com qualquer formato de *frame* (pacote) que estiver em uso na rede local à qual o computador está

ligado. Esse encapsulamento requer a inclusão de um endereço de rede local ou endereço físico no *frame*.

O segundo aspecto é necessário porque a Internet consiste de um número de redes locais interconectadas por um ou mais roteadores. Tais roteadores, também conhecidos como *gateways*, algumas vezes tem conexões físicas ou portas de acesso à mais de uma rede. A identificação do roteador e porta apropriados para onde um pacote IP particular deve ser enviado é chamada de roteamento, que também envolve a troca de informações entre os roteadores de forma padronizada.

O terceiro aspecto envolve a tradução do endereço de rede. Essa tradução é feita por um sistema conhecido como DNS (*Domain Name Service*) [6]. O DNS é um protocolo que traduz nomes de domínios em endereços IP. Sendo os nomes de domínio alfabéticos, a sua memorização é mais fácil. A Internet, contudo, baseia-se em endereços IP. Por essa razão, toda a vez que se usa um nome de domínio, o DNS deve traduzir esse nome em um endereço IP correspondente. Por exemplo, o nome de domínio **www.cbpf.br** deve ser traduzido para **152.84.253.64** [9].

2.3. Determinação do Endereço Físico

Se um computador quer transmitir dados através de uma rede TCP/IP, é necessário encapsulá-los em uma forma apropriada para o meio físico da rede a qual o computador está ligado. Para que a transmissão desse *frame* seja completada, é preciso determinar o endereço físico do computador de destino, que pode ser obtido usando-se uma tabela, configurada como um arquivo que é lido pela memória do computador toda vez que este for inicializado, que relacione os endereços IP dos computadores conectados à rede com os seus respectivos endereços físicos. Todos os computadores da rede contém essa tabela.

Na prática, entretanto, os computadores criam essa tabela através da utilização de um protocolo de comunicação conhecido como ARP (*Address Resolution Protocol*). Esse protocolo, definido pela RFC 826, é usado para associar endereços IP com endereços físicos (endereços de MAC). A tabela criada e atualizada por esse protocolo, que associa esses dois tipos de endereço, é chamada de ARP *cache*.

Quando um *host* quer comunicar-se com outro, mas não tem o seu endereço físico, ele envia uma mensagem ARP, encapsulada em um *frame*, contendo os endereços IP e físico de origem na rede por *broadcast*; isto é, essa mensagem é transmitida para todos os computadores da rede. Os computadores então armazenam esse endereço físico de origem, e o *host* de destino responde a esse *broadcast* enviando o seu endereço físico para o *host* que originalmente transmitiu o pacote ARP. O ARP *cache* é criado a partir do armazenamento dos MAC's por parte dos *hosts* da rede, sendo que os seus registros expiram após um determinado período de tempo, geralmente alguns minutos.

3. *Firewall*

Um *firewall* é definido como um sistema designado para prevenir acessos não-autorizados à redes de computadores. Os *firewalls* podem ser implementados tanto em hardware quanto em software, ou ainda em uma combinação de ambos. Esse sistema é utilizado freqüentemente em redes privadas conectadas com a Internet, especialmente as intranets, para evitar que usuários não-autorizados tenham acesso à elas. Esse controle é feito através da checagem das mensagens que entram e saem da intranet. Essas mensagens passam pelo *firewall*, que as examina,

uma a uma, e bloqueia aquelas que não obedecem aos critérios de segurança especificados pelo administrador da rede [10].

Como exemplo, consideremos a figura 3 mostrada abaixo. O computador externo à LAN pode conectar-se a qualquer um dos seus dispositivos através do roteador, não havendo a princípio qualquer tipo de controle de acesso às máquinas e às informações armazenadas nessa rede. Dessa forma, pessoas não-autorizadas podem ter acesso a esses dados, podendo então lê-los, modificá-los ou até mesmo apagá-los remotamente.
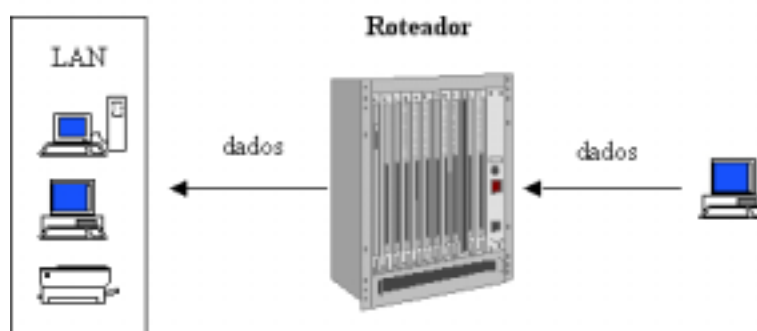
Figura 3: O roteador permite o tráfego de informações entre computadores externos à LAN e os seus dispositivos, a princípio sem qualquer tipo de segurança.

Para evitar isso, utiliza-se um *firewall*. Com esse sistema, os dados transmitidos pelo computador externo continuam sendo trafegados pela rede, desde que sejam permitidos pelos filtros do *firewall*. Se não forem, os dados são bloqueados pelo *firewall*, que envia então uma mensagem ao computador de origem dizendo que a transmissão não foi completada, protegendo assim a LAN de eventuais ataques, como mostram as figuras 4 e 5.
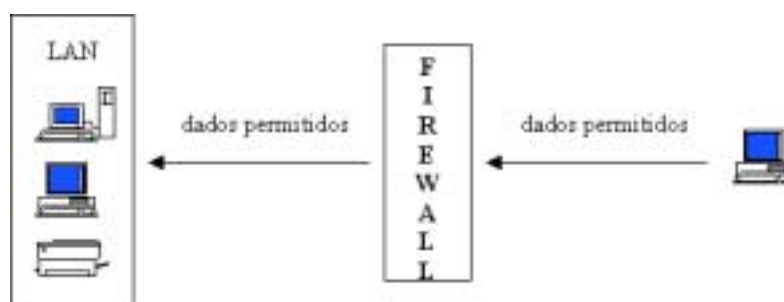
Figura 4: Com a implementação do *firewall*, os dados continuam a ser transmitidos, desde que sejam permitidos.
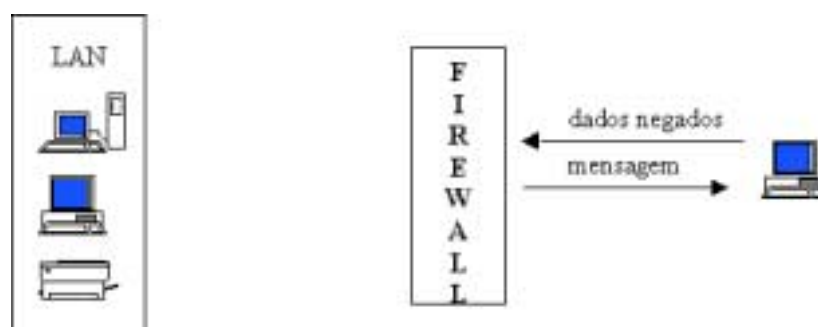
Figura 5: Caso a transmissão dos dados seja negada pelo *firewall*, uma mensagem é transmitida para o computador externo à LAN avisando que a transmissão não foi completada.

Existem diversos tipos de técnicas de *firewalls*, que serão discutidas em detalhes nas seções a seguir.

3.1. Filtros de Pacotes

Nessa técnica de *firewall*, temos que filtros (linhas de comando configuradas no roteador) checam cada pacote que entre e sai da rede local, aceitando-os ou bloqueando-os de acordo com as regras definidas pelo administrador da rede. Esse controle de acesso é feito através da análise dos endereços IP de origem e destino de cada pacote e das portas UDP e TCP utilizadas pela rede. O administrador elabora uma lista dos dispositivos e serviços oferecidos que estão autorizados a transmitir dados nos sentidos de transmissão possíveis. Essa lista é então usada para filtrar os pacotes que tentam atravessar o *firewall*. Um exemplo de política de filtragem de pacotes seria: permitir o tráfego *full-duplex* de pacotes carregando mensagens de SMTP e DNS, tráfego Telnet só para pacotes saindo da rede e bloquear todos os outros tipos de tráfego.

A filtragem de pacotes é uma técnica de *firewall* bastante efetiva e transparente para os usuários, mas de difícil configuração. Além disso, a abordagem baseada em filtragem não fornece uma granularidade muito fina de controle de acesso, uma vez que o acesso é controlado com base nas máquinas de origem e de destino dos pacotes, e é vulnerável a certos tipos de ataques, que estão listados abaixo [1][10]:

3.1.1. IP *Spoofing*

Esse ataque consiste no ganho de acesso não-autorizado a computadores de uma rede privada. Mensagens são enviadas para o computador que será invadido com endereços IP que indicam que essas mensagens estão vindo de um *host* interno da rede. Para isso, deve-se primeiramente usar uma variedade de técnicas que permitam achar um endereço IP de um *host* da rede, e então modificar os cabeçalhos dos pacotes de forma que eles pareçam estar sendo transmitidos por esse *host*. Assim, espera-se que o uso desse endereço IP falso permita o acesso à sistemas que tenham uma política de segurança simples, baseada somente na checagem dos endereços de destino, onde os pacotes enviados por *hosts* internos da rede são aceitos enquanto pacotes enviados por qualquer outro *host* são descartados. O IP *spoofing* pode ser evitado descartando-se pacotes com endereços de origem internos que venham de uma das saídas de uma interface do roteador da rede [13][14].

3.1.2. Ataque *Source Routing*

Em um ataque do tipo *source routing*, a estação de origem especifica a rota que um pacote deve seguir ao ser transmitido pela Internet. Esse tipo de ataque é designado para fugir de medidas de segurança e assim fazer com que o pacote siga uma rota não esperada até o seu destino. Um ataque *source routing* pode ser evitado simplesmente descartando-se todos os pacotes que contenham em seus cabeçalhos a opção *source route* [13].

3.1.3. Ataque *Tiny Fragment*

Para esse tipo de ataque, utiliza-se o aspecto de fragmentação de pacotes IP’s para criar fragmentos extremamente pequenos e assim forçar o cabeçalho TCP de informação a ser um fragmento de pacote separado. O ataque *tiny fragment* é designado para evitar as regras de filtragem

da política de segurança da rede; espera-se que a filtragem implementada no roteador examine somente o primeiro fragmento do pacote transmitido, permitindo assim a passagem dos restantes. Esse ataque pode ser evitado descartando-se aqueles pacotes em que o tipo de protocolo é o TCP e o parâmetro IP *FragmentOffset*, especificado no cabeçalho, é igual à 1 [13].

3.2. *Gateways* de aplicação

Firewalls na Internet são muitas vezes considerados como *gateways* de segurança que controlam o acesso a uma rede. Na linguagem dos *firewalls*, um *gateway* é um dispositivo que oferece serviços de transmissão de dados entre duas redes. Um *firewall* pode ser mais do que um filtro no roteador, como é o *gateway* controlado. Nesse caso, o tráfego passa pelos filtros do *gateway* ao invés de ser transmitido diretamente na rede. Após a checagem dos dados, o *gateway* então os transmite para uma outra rede ou para o *gateway* que estiver conectado à ela [12].

O *gateway* de aplicação atua na camada de aplicação. Esse tipo de *firewall* aplica mecanismos de segurança em aplicações específicas, como por exemplo servidores FTP e servidores Telnet. Devido a sua grande flexibilidade, o *gateway* de aplicação pode fornecer um maior grau de proteção do que o filtro de pacote. Contudo, apesar de eficiente, essa técnica pode impor uma degradação na performance da rede.

Para melhor compreender essa técnica de *firewall*, consideremos o seguinte exemplo: um *gateway* FTP é configurado para restringir as operações de transferência de arquivos que estejam localizados no *bastion host* (*gateway* do *firewall* que pode ser acessado a partir da rede externa). Dessa forma, os usuários externos só podem ter acesso aos arquivos disponibilizados nessa máquina, o *bastion host*. Além disso, a aplicação FTP original pode ser modificada para limitar a transferência de arquivos da rede interna para a rede externa a usuários autorizados, e ainda com limites para o volume de informação que pode ser transmitida, dificultando assim ataques externos [1][10]. A figura 6 mostra os componentes que compõem esse tipo de *firewall*.
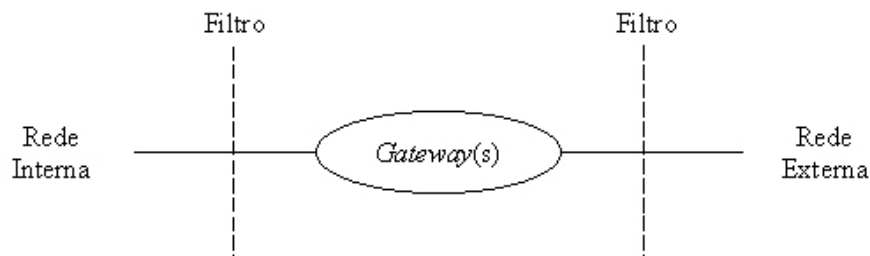
Figura 6: Componentes de um *gateway* de aplicação.

3.3. *Gateways* de circuito

Esse *firewall* aplica mecanismos de segurança quando uma conexão TCP ou UDP é estabelecida. Ele atua como intermediário de conexões FTP, funcionando como um TCP modificado. Para permitir a transmissão dos dados através desse tipo de *firewall*, o usuário de origem conecta-se a uma porta TCP no *gateway*, que por sua vez conecta-se, usando outra conexão TCP, ao usuário de destino. Um circuito é então formado por uma conexão TCP na rede interna e outra na rede externa, estando ambas associadas pelo *gateway* de circuito. O processo que implementa esse tipo de *gateway* atua repassando bytes de uma conexão para outra, fechando então o circuito. Para que esse circuito seja estabelecido, o usuário de origem deve fazer uma solicitação ao *gateway* no *firewall*, passando a máquina e o serviço de destino como parâmetros. Com isso, o

gateway estabelece o circuito ou, em caso contrário, retorna um código informando o motivo do não estabelecimento. Uma vez que a conexão tenha sido estabelecida, os pacotes de dados podem trafegar entre os *hosts* da rede sem checagens adicionais. É importante notar que é necessário que o usuário de origem utilize um protocolo simples para comunicar-se com o *gateway*, sendo esse protocolo um bom local para implementar, por exemplo, um mecanismo de autenticação [1][10].

3.4. Servidores *Proxy*

É um servidor que é implementado entre a aplicação de um cliente (arquitetura cliente/servidor), como por exemplo um navegador *Web*, e um servidor real. O servidor *proxy* intercepta todos os pedidos requeridos ao servidor real e verifica se ele mesmo pode executar esses pedidos. Se não for possível, ele então transmite o pedido para o servidor real. O servidor *proxy* tem dois propósitos básicos [15]:

Melhoria de Performance: Os servidores *proxy* podem melhorar sensivelmente a performance de grupos de usuários, uma vez que eles armazenam os resultados de todos os pedidos feitos pelos usuários ao servidor real durante um determinado período de tempo. Consideremos por exemplo o caso em que ambos os usuários X e Y de um grupo de usuários qualquer de uma rede acessam a *World Wide Web* através de um servidor *proxy*. Primeiramente, o usuário X visita uma certa página *Web*, que será chamada aqui de página 1. Algum tempo depois, o usuário Y tenta visitar essa mesma página. Ao invés de transmitir o pedido para o servidor *Web* onde a página 1 reside, o que pode ser uma operação que consuma muito tempo, o servidor *proxy* simplesmente retorna a página 1 que já foi acessada pelo usuário X. Considerando-se o fato do servidor *proxy* estar na maioria das vezes na mesma rede que o usuário, essa operação aqui descrita torna-se muito mais rápida [15].

Filtragem de Pedidos *(Firewall)*: Os servidores *proxy* também podem ser usados para filtrar os pedidos requeridos ao servidor real da rede. Eles interceptam todas as mensagens que entram e saem da rede, escondendo assim de forma efetiva os seus endereços. Através da utilização de um servidor desse tipo, uma companhia pode prevenir que os seus empregados tenham acesso a algum grupo específico de *Web sites* [10][15].

3.5. Comentários finais

Na prática, muitos *firewalls* usam duas ou mais dessa técnicas aqui apresentadas simultaneamente. Ainda, temos que, para uma maior segurança, os dados transmitidos pela rede também podem ser encriptados. Como exemplo prático, temos que um *firewall* pode ser implementado através da utilização de listas de acesso (*access lists*), assunto do próximo item.

4. Access Lists

Nesse item será estudado o conceito de *access list*, uma ferramenta disponível em alguns roteadores, como por exemplo o roteador Cisco IOS, que é utilizada para garantir a integridade dos dados que são trafegados e armazenados em uma rede local. Abaixo, segue as razões para o uso das *access lists*, os seus conceitos básicos, como funcionam, como são implementadas e alguns exemplos práticos.

4.1. Para que usar *Access Lists*?

Basicamente, o comando *access list* é utilizado para controlar o tráfego de dados em uma rede local através de testes de pacotes. Essas listas oferecem uma poderosa ferramenta para o controle da rede: a flexibilidade para filtrar o fluxo de pacotes que são transmitidos pelas entradas e saídas das interfaces do roteador. Esse comando ajuda a proteger as expansões dos recursos de rede, sem impedir o fluxo da comunicação dos dados, diferenciando o tráfego desses dados em categorias que são permitidas ou não. Esse controle na transmissão da informação baseia-se em aspectos determinados pelo administrador da rede [3].

4.2. O que são *Access Lists*?

Access lists são linhas de comando que especificam condições de controle determinadas pelo administrador da rede. Baseado nessas determinações, o roteador lidará com o tráfego de pacotes de forma diferenciada, de acordo com o tipo de dado que está sendo trafegado. Existem dois tipos básicos de *access lists*, que estão listados abaixo:

Standard access lists**:** *Standard access lists* para IP checam somente o endereço de origem dos pacotes que podem ser roteados. De acordo com o resultado dessa checagem, o roteador pode permitir ou não a transmissão de um grupo inteiro de protocolos, baseando-se no endereço de rede, sub-rede ou do *host*. Por exemplo; pacotes chegando através da entrada de uma interface do roteador são checados pelos seus endereços. Se a permissão for dada pela *access list*, o roteador transmite os pacotes através da saída da interface que estiver associada a essa *access list*. Por outro lado, se a permissão para a transmissão não for dada, os pacotes serão desprezados.

Extended access lists**:** *Extended access lists* checam tanto o endereço de destino quanto o de origem dos pacotes. Esse tipo de lista também pode checar outros tipos de parâmetros, como por exemplo protocolos específicos e números de porta. Com isso, os administradores de rede tem maior flexibilidade para descrever qual tipo de filtragem a *access list* deve fazer. Tem-se que um pacote tem o seu tráfego permitido ou não com base na sua origem e no seu destino. Além disso, temos também que as *extended access lists* controlam o fluxo de pacotes com maior granularidade; isto é, pode-se permitir por exemplo o tráfego de *e-mails* da entrada de uma interface para a sua saída, e ao mesmo tempo proibir o acesso remoto e a transferência de arquivos entre elas [3].

4.3. Comandos das *Access Lists*

Na prática, a implementação e a interpretação das *access lists* podem ser complicadas. Entretanto, o entendimento dos comandos de configuração dessas listas pode ser simplificado reduzindo-os em dois passos básicos, que estão esquematizados abaixo:

Passo 1: Configurar os parâmetros de teste para a linha de comando da *access list* (que pode ser uma entre várias linhas de comando que formam a *access list*).

```
User Access Verification
Password:
router>enable
Password:
router#config t
Enter configuration commands, one per line. End with CNTL/Z
router(config)#access-list access-list-number {permit | deny} {test conditions}
```

A última linha de comando, mostrada no exemplo acima, identifica a *access list*, que é geralmente representada por um número (termo ***access-list-number***). Esse número indica que tipo de *access list* foi implementada. O termo **permit | deny** indica como o roteador deverá lidar com os pacotes que serão checados pelos testes de condições, que são especificados pelo último termo da linha de comando, o **test conditions**. Na maioria das vezes, o termo **permit** significa que o pacote tem permissão para trafegar através de uma ou mais interfaces do roteador.

Passo 2: Abilitar uma interface do roteador para fazer parte do grupo que usa a *access list* especificada.

```
User Access Verification
Password:
router>enable
Password:
router#config t
Enter configuration commands, one per line. End with CNTL/Z
router(config)#access-list access-list-number {permit | deny} {test conditions}
router(config)#int router interface
router(config-if)#{protocol} access-group access-list-number
```

O comando *access list* usa um comando de interface. Todas as linhas de comando da *access list* que são identificadas pelo seu número são associadas a uma ou mais interfaces do roteador, que é identificada através do comando **int *router interface***, onde ***router interface*** indica qual a interface utilizada. Com isso, qualquer pacote de dados que passe pelos testes de condições da *access list* tem a permissão de usar qualquer interface que faça parte do *access group* de interfaces [3].

4.4. Identificação de *Access Lists*

O comando *access list* pode controlar vários protocolos em um roteador. A tabela abaixo mostra os tipos de *access lists* e os seus respectivos intervalos de números de identificação [4].

Tipo de *Access List*	Intervalo do Número de Identificação
IP *standard*	1 – 99
IP *extended*	100 – 199
Bridge type-code	200 – 299
DECnet *standard* e *extended*	300 – 399
XNS *standard*	400 – 499
XNS *extended*	500 – 599
AppleTalk zone	600 – 699
Bridge MAC	700 – 799

IPX *standard*	800 – 899
IPX *extended*	900 – 999
IPX SAP	1000 – 1099
Bridge extended	1100 – 1199

De todos os tipos de *access list* mostrados na tabela acima, apenas os dois primeiros tipos serão tratados nessa Nota Técnica. Nos próximos itens, serão vistas as suas respectivas configurações.

4.5. Configuração de IP *Standard Access Lists*

User Access Verification
Password:
router>enable
Password:
router#config t
Enter configuration commands, one per line. End with CNTL/Z
router(config)#access-list *access-list-number* {permit | deny} *source* [*source-mask*]
router(config)#int *router interface*
router(config-if)#ip access-group *access-list-number* {in | out}

O comando *access-list* cria uma entrada em uma lista *standard* de filtragem de pacotes. A seguir, encontramos uma tabela com as respectivas descrições dos termos que constituem esse comando.

Comando *access* list	Descrição
access-list-number	Identifica a lista à qual a entrada pertence; um número de 1 até 99.
permit | deny	Indica se a entrada irá permitir ou bloquear o tráfego do pacote especificado.
source	Identifica o endereço IP de origem.
source-mask	Identifica quais *bits* do campo de endereço devem ser checados. Há um 1 nas posições que indicam *bits* desprezados, e um 0 em qualquer posição em que o *bit* deve ser obrigatoriamente checado.

O comando **ip access-group** liga o objeto *access list* existente à saída de uma interface do roteador. Somente um objeto *access list* por porta, protocolo e direção é permitida.

Para retirar um objeto *access list*, primeiramente entre com o comando *no access-group* com todo o seu grupo de parâmetros, e depois entre com o comando *no access-list* com todo o seu grupo de parâmetros [3].

Comando ip *access-group*	**Descrição**
access-list-number	Indica o número da *access list* que deve ser ligada a interface desejada.
in | out	Seleciona se a *access list* é abilitada para a entrada ou saída de uma interface. Se in ou out não forem especificadas, out é o padrão.

4.6. Configuração de IP *Extended Access Lists*

User Access Verification
Password:
router>enable
Password:
router#config t
Enter configuration commands, one per line. End with CNTL/Z
router(config)#access-list *access-list-number* {permit | deny} *protocol source source-mask destination destination-mask* [*operator operand*] [established]
router(config)#int *router interface*
router(config-if)#ip access-group *access-list-number* {in | out}

A seguir, temos as tabelas com as descrições dos termos que constituem os comandos usados para esse tipo de *access list* [3].

Comando access-list	**Descrição**
access-list-number	Identifica a lista usando um número entre 100 e 199
permit | deny	Indica se a entrada permite ou bloqueia o endereço especificado.
protocol	IP, TCP, UDP, ICMP, GRE, IGRP.
source e *destination*	Identifica os endereço IP de origem e de destino.
source-mask e *destination-mask*	Máscaras. Os 0's indicam os bits que devem serchecados, enquanto os 1's indicam os bits que são desprezados.
operator e *operand*	lt, gt, eq, neq (*less than, greater than, equal, not equal*) e um número de porta.

Comando ip *access-group*	**Descrição**
access-list-number	Indica o número da *access list* que deve ser ligada à interface desejada.
in | out	Seleciona se a *access list* é abilitada para a entrada ou saída de uma interface. Se **in** ou **out** não forem especificadas , **out** é o padrão.

4.7. Exemplos de *Access Lists*

Nesse item, serão mostrados um exemplo prático da utilização de uma *standard access lists* e o de uma *extended access list*, de forma a mostrar com maior clareza a utilidade dessas ferramentas em um ambiente de rede.

4.7.1. Standard Access Lists

Exemplo: Bloquear a transmissão de uma rede específica.
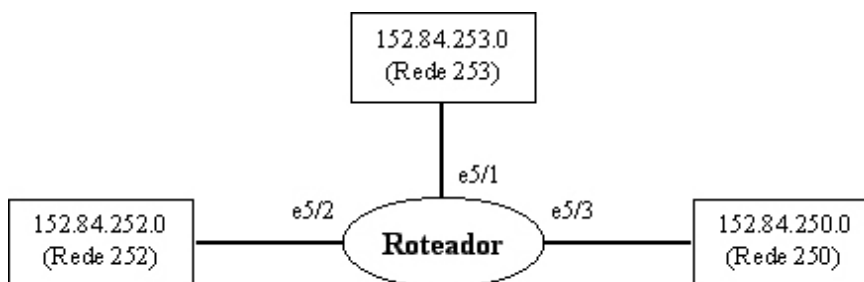
Figura 7: Roteador interligando diferentes redes através de suas interfaces.

```
User Access Verification
Password:
router>enable
Password:
router#config t
Enter configuration commands, one per line. End with CNTL/Z
router(config)#access-list 1 permit  152.84.250.0  0.0.0.255
router(config)#access-list 1 permit  0.0.0.0  255.255.255.255
router(config)#int e5/2
router(config-if)#ip access-group 1
```

Nesse exemplo, a *access list* está configurada para bloquear o tráfego transmitido pela rede 250 (152.84.250.0) e permitir o tráfego transmitido pelas redes 252 e 253 (152.84.252.0 e 152.84.253.0), através da interface e5/2.

4.7.2. Extended Access Lists

Exemplo: Bloquear transmissão por FTP pela interface e5/1
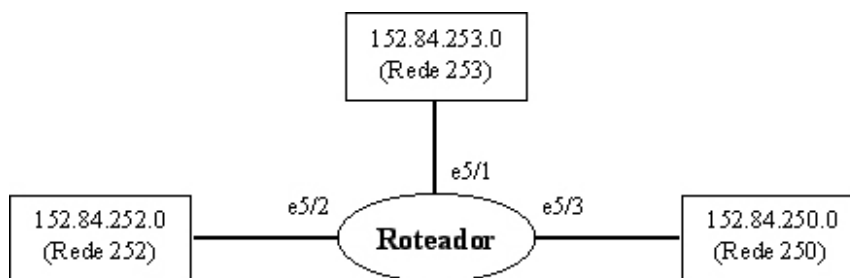
Figura 8: Roteador interligando diferentes redes através de suas interfaces.

```
User Access Verification
Password:
router>enable
Password:
router#config t
Enter configuration commands, one per line. End with CNTL/Z
router(config)#access-list 101 deny tcp  152.84.250.0  0.0.0.255  152.84.252.0  0.0.0.255  eq 21
router(config)#access-list 101 deny tcp  152.84.250.0  0.0.0.255  152.84.252.0  0.0.0.255  eq 20
router(config)#access-list 101 permit ip  152.84.250.0  0.0.0.255  0.0.0.0  255.255.255.255
router(config)#int e5/2
router(config-if)#ip access-group 101
```

Nesse exemplo, a *access list* bloquea o tráfego de pacotes FTP transmitidos pela rede 250 (152.84.250.0), permitindo que o tráfego restante dessa rede seja transmitido para todas as outras redes (152.84.252.0 e 152.84.253.0), tudo isso pela interface e5/2.

5. Segurança em Redes de Computadores

A segurança da rede local do CBPF, como a segurança de qualquer LAN, está relacionada à necessidade de proteção contra acessos não autorizados, manipulação dos dados armazenados na rede, assim como a sua integridade, e utilização não autorizada de computadores ou de seus respectivos dispositivos periféricos. Essa necessidade de proteção deve ser definida a partir das possíveis ameaças e riscos que a rede sofre, além dos objetivos traçados pela instituição, sendo tudo isso formalizado nos termos de uma política de segurança. Dessa forma, procura-se evitar que pessoas não-autorizadas tenham acesso a informações particulares de qualquer usuário da rede. Nos próximos itens, estudaremos um pouco mais sobre política de segurança e conheceremos alguns desses mecanismos aplicados a ambientes de comunicação de dados [1].

5.1. Política de Segurança

Uma política de segurança é definida como sendo um conjunto de leis, regras e práticas que definem como uma empresa ou instituição gerencia e protege seus recursos e transmite os seus dados. Um sistema de comunicação de dados pode ser considerado seguro quando garante o cumprimento dessa política, que deve incluir regras detalhadas definindo como as informações e recursos oferecidos pela rede devem ser manipulados.

Uma política de segurança é implementada baseando-se na aplicação de regras que controlem o acesso aos dados e recursos que são trafegados através da rede; isto é, define-se o que é e o que não é permitido em termos de segurança, durante a operação de um dado aplicativo ou recurso da rede, através da definição do nível de acesso autorizado para os usuários que utilizam-se do sistema de comunicação de dados. Com base na natureza da autorização que é dada ao usuário, pode-se dividir em dois os tipos de política de segurança existentes: uma baseada em regras, onde os dados e recursos da rede são marcados com rótulos de segurança apropriados que definem o nível de autorização do usuário que os está controlando; e uma outra baseada em identidade. Nesse último tipo, temos que o administrador da rede pode especificar explicitamente os tipos de acesso que os usuários da rede podem ter às informações e recursos que estão sob seu controle [1].

5.2. Mecanismos de Segurança

Uma política de segurança pode ser implementada com a utilização de vários mecanismos. Abaixo, temos alguns dos mais importantes mecanismos de segurança utilizados em redes de computadores.

5.2.1. Criptografia

Em meios de comunicação onde não é possível impedir que o fluxo de pacote de dados seja interceptado, podendo as informações serem lidas ou até modificadas, é necessária a criptografia. Nesse mecanismo, utiliza-se um método que modifique o texto original da mensagem transmitida, gerando um texto criptografado na origem, através de um processo de codificação definido por um método de criptografia. O pacote é então transmitido e, ao chegar no destino, ocorre o processo inverso; isto é, o método de criptografia é aplicado agora para decodificar a mensagem, transformando-a na mensagem original.

Contudo, toda a vez que o método utilizado é descoberto, quebrando-se o código de criptografia, é necessário substituí-lo por um outro diferente, o que acarreta no desenvolvimento de novos procedimentos para a implementação desse novo método, treinamento do pessoal envolvido, etc. Com o intuito de evitar tal problema, criou-se um novo mecanismo de criptografia, representado na figura 9 mostrada abaixo. Nesse novo modelo, um texto criptografado gerado a partir do texto normal varia de acordo com uma chave de codificação utilizada para o mesmo método de criptografia. Isto é, para uma mesma mensagem original e um mesmo método de criptografia, chaves diferentes produzem textos criptografados diferentes. Dessa forma, não adianta conhecer o método de criptografia para recuperar a mensagem original, porque, para recuperá-la corretamente, é necessário tanto o texto criptografado quanto a chave de decodificação utilizada [1].
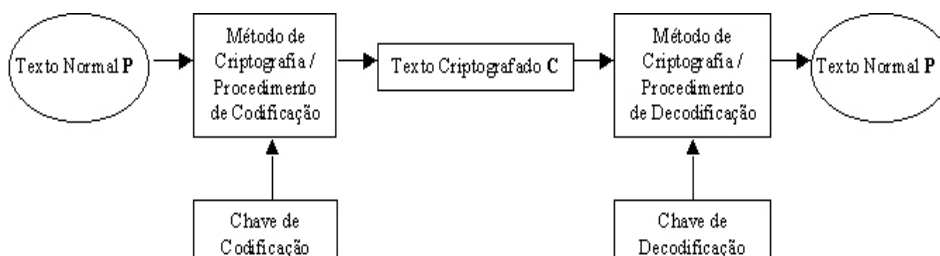
Figura 9: Método de criptografia utilizando chaves.

5.2.2. Integridade de Dados

Os mecanismos de controle de integridade de dados atuam em dois níveis: controle da integridade de pacotes isolados e controle da integridade de uma conexão, isto é, dos pacotes e da seqüência de transmissão.

Em relação ao primeiro nível, tem-se que técnicas de detecção de modificações, que são normalmente associadas com a detecção de erros em bits, pacotes ou erros de seqüência introduzidos por enlaces e redes de comunicação, são usadas para garantir a integridade dos dados trafegados em uma rede. Contudo, se os cabeçalhos dos pacotes de dados não forem devidamente

protegidos contra possíveis modificações, pode-se contornar a verificação, desde que sejam conhecidas essas técnicas. Portanto, para garantir a integridade é necessário manter confidenciais e íntegras as informações de controle que são usadas na detecção de modificações.

Já para controlar modificações na seqüência de pacotes transmitidos em uma conexão, são necessárias técnicas que garantam a integridade desses pacotes, de forma a garantir que as informações de controle não sejam corrompidas, em conjunto com informações de controle de seqüência. Esses cuidados, apesar de não evitarem a modificação da cadeia de pacotes, garantem a detecção e notificação dos ataques [1].

5.2.3. Controle de Acesso

Esse mecanismo de segurança é utilizado para garantir que o acesso a um recurso de rede qualquer seja limitado a usuários devidamente autorizados pelo administrador do sistema. Como técnicas utilizadas, tem-se a utilização de listas ou matrizes de controles de acesso, que associam recursos a usuários autorizados; ou senhas e *tokens* associadas aos recursos, cuja posse determina os direitos de acesso do usuário que as possui.

Como exemplo da utilização de *tokens* para controlar o acesso aos recursos de uma rede, consideremos o método de controle de congestionamento de tráfego conhecido como **controle isorrítmico**. Nesse método, existem permissões, que são os *tokens*, que circulam pela rede. Sempre que um *host* deseja transmitir um novo pacote pela rede, ele primeiramente deve capturar uma dessas permissões e destruí-la, sendo que essa permissão destruída é regenerada pelo *host* que recebe o pacote no destino. Contudo, esse método apresenta um problema: a distribuição das permissões depende das aplicações na rede e o próprio tráfego aleatório desses *tokens* causa um tráfego extra na rede, diminuindo assim a sua performance. Ainda, tem-se que a perda de uma permissão devido a uma falha qualquer na rede deve ser recuperada, de forma a evitar que a sua capacidade de transporte seja reduzida [1].

5.2.4. Controle de roteamento

Esse mecanismo garante a transmissão de informação através de rotas fisicamente seguras, cujos canais de comunicação forneçam os níveis apropriados de proteção. Essa garantia se deve ao controle do roteamento de pacotes de dados. Através desse controle, rotas preferenciais (ou obrigatórias) para a transferência de dados são especificadas pelo administrador do sistema [1].

5.2.5. Comentários finais

Além desses mecanismos de segurança mencionados, muitos outros são encontrados. Em alguns casos particulares, a política de segurança aplicada é baseada na implementação de *firewalls*, já estudados anteriormente nessa Nota Técnica.

6. Conclusão

Ao final desse trabalho, podemos concluir que o roteador apresenta uma importância muito grande dentro de uma topologia de rede, uma vez que esse equipamento permite, além da gerência e do controle de acesso às informações, o compartilhamento de recursos e serviços entre

redes geograficamente dispersas. Como conseqüência, é essencial que o administrador da rede tenha total domínio sobre os recursos do roteador que a gerencia, assim como o entendimento da tecnologia desse dispositivo, pois assim será possível garantir uma maior qualidade na sua performance.

Ainda, também podemos observar como é importante a implementação de uma política de segurança adequada que garanta a integridade das informações armazenadas na rede e dos dispositivos que a compõem, protegendo assim esses dados e máquinas de possíveis ataques externos, que poderiam causar sérios danos a estrutura administrativa e técnica de uma empresa ou organização.

Referências

Em relação às referências de páginas *web*, é importante ressaltarmos que páginas da Internet são altamente dinâmicas, podendo mudar desde parte do seu conteúdo até o próprio endereço. Devido a isto, existe a possibilidade de que, ao consultar alguma página, essa não mais exista ou então tenha outras informações.

[1] "Redes de Computadores – Das LAN's, MAN's e WAN's às Redes ATM – Segunda edição" / Luiz Fernando Gomes Soares, Guido Lemos e Sérgio Colcher.

[2] "Redes de Computadores" / Nota Técnica escrita por Leonardo Ferreira Carneiro, Nilton Costa Braga e Nilton Alves Jr. em Outubro de 1998.

[3] "*Introduction to Cisco Router Configuration*: *Student Guide*" / Manual da *Cisco Systems Inc*.

[4] "*Advanced to Cisco Router Configuration*: *Student Guide*" / Manual da *Cisco Systems Inc*.

[5] "*Router*", http://webopedia.internet.com/TERM/r/router.html.

[6] "*Routing in the Internet*", http://www.scit.wlv.ac.uk/~jpjb/comms/iproute.html.

[7] "*Header*", http://webopedia.internet.com/TERM/h/header.html.

[8] "ICMP", http://webopedia.internet.com/TERM/I/ICMP.html.

[9] "DNS", http://webopedia.internet.com/TERM/D/DNS.html.

[10] "*firewall*", http://webopedia.internet.com/TERM/f/firewall.html.

[11] "*The Need for Firewalls*", http://www.icsa.net/fwbg/chap_2.html.

[12] "*What is a Firewall?*", http://www.icsa.net/fwbg/chap_3.html.

[13] "*Internet Firewalls and Security – A Technology Overview*" / Escrito por *Chuck Semeria*http://www.3com.com/nsc/500619.html

[14] "IP *spoofing*", http://webopedia.internet.com/TERM/I/IP_spoofing.html.

[15] "*proxy server*", http://webopedia.internet.com/TERM/p/proxy_server.html.

[16] "*Routing Information Protocol*", http://webopedia.internet.com/TERM/R/Routing_Information_Protocol.html.

[17] "*Networking: A Primer*", http://ww.baynetworks.com/products/Papers/wp-primer.html.

[18] "Internetworking", http://webopedia.internet.com/TERM/i/internetworking.html.

[19] "ARP", http://webopedia.internet.com/TERM/A/ARP.html.